CLAIMS

1. **(Currently Amended)** A method comprising:

receiving an input of data, the input data conforming to a query language used by a filter engine comprising two or more filter sub-engines, wherein at least one filter sub-engine is a general filter sub-engine and at least one filter sub-engine is an optimized filter sub-engine, and wherein the query language is based on eXtensible Markup Language (XML);

determining whether the input data conforms to a grammar associated with the optimized filter sub-engine, wherein the optimized filter sub-engine is configured to handle only a subset of the query language handled by the general filter sub-engine;

in an event the determining indicates the input data conforms to the grammar associated with the optimized filter sub-engine:

determining whether the input data can be processed by the optimized filter sub-engine, the determining comprising identifying if the input data comprises a subset of the query language; and

directing the input data to the optimized filter sub-engine for processing;

in an event the determining indicates that the input data cannot be processed by the optimized filter sub-engine:

determining whether the input data can be processed by a second optimized filter sub-engine, wherein the second optimized filter sub-engine is configured to handle only a subset of the query language, and wherein the subset of the query language that the second optimized filter sub-engine is configured to handle excludes the subset of the query language that the first optimized filter sub-engine is configured to handle; and

directing the input data to the second optimized filter sub-engine for processing;

in an event the determining indicates that the input cannot be processed by the second optimized filter sub-engine, directing the input to the general filter sub-engine for processing, wherein the general filter sub-engine is configured to handle all aspects of the query language; and

processing the input to derive a result.

2. (Canceled)

3. (Canceled)

4. (Canceled)

5. (Canceled)

6. (Previously Presented) The method as recited in claim 1, the method further comprising:

parsing the input to identify first and second sub-expressions;

determining whether the first sub-expression can be processed by the optimized filter sub-engine;

in an event the first sub-expression can be processed by the optimized filter sub-engine, ~~then~~ directing the first sub-expression to the optimized filter sub-engine for processing;

in an event the first sub-expression cannot be processed by the optimized filter sub-engine, directing the first sub-expression to the general filter sub-engine for processing;

determining whether the second sub-expression can be processed by the optimized filter sub-engine;

in an event the second sub-expression can be processed by the optimized filter sub-engine, directing the second sub-expression to the optimized filter sub-engine for processing; and

in an event the second sub-expression cannot be processed by the optimized filter sub-engine, directing the second sub-expression to the general filter sub-engine for processing.

7. **(Original)** The method as recited in claim 6, further comprising:

obtaining a result of the processing of the first sub-expression; and

processing the second sub-expression only if the result of the first sub-expression is true.

8. **(Currently Amended)** A filter engine system comprising:

a processor coupled to a memory, the memory configured with instructions for implementing:

an optimized filter sub-engine configured to accept an input that conforms to a language and process the input against a filter table associated with the optimized filter sub-engine, wherein the optimized filter sub-engine is configured to process only a subset of terms of the language, wherein the subset of terms of the language does not include all terms of the language, and wherein the language comprises a query language based on eXtensible Markup Language (XML);

a general filter sub-engine configured to accept the input and process the input against a filter table associated with the general filter sub-engine, wherein the general filter sub-engine is configured to process all terms of the input language; and

an analyzer configured to determine whether the input can be processed by the optimized filter sub-engine and, if so, direct the input to the optimized filter sub-engine for processing or, if not, direct the input to the general filter sub-engine for processing.

9. **(Currently Amended)** The filter engine system as recited in claim 8, wherein the analyzer is further configured to analyze a new filter added to the filter engine and to determine an appropriate filter sub-engine with which to associate the new filter.

10. **(Currently Amended)** The filter engine system as recited in claim 8, wherein the language ~~is XPath~~ additionally comprises XML Path Language (XPath).

11. **(Currently Amended)** The filter engine system as recited in claim 8, wherein the analyzer is further configured to determine whether the optimized filter sub-engine can process the input by comparing the input to a grammar associated with the optimized filter sub-engine and determining whether the input consists of terms that are compatible with the grammar.

12. **(Currently Amended)** The filter engine system as recited in claim 8, further comprising a sub-expression module that is configured to perform acts comprising:

determine whether the input consists of different sub-expressions;

in an event the input consists of different sub-expressions, directing each of the different sub-expressions contained in the input to the analyzer, wherein the analyzer is further configured to determine whether each of the different sub-expressions can be processed by the optimized filter sub-engine and to direct each of the different sub-expressions to an appropriate filter sub-engine for processing.

13. (Currently Amended) The filter engine system as recited in claim 12, wherein a first of the different sub-expressions is directed to the optimized filter sub-engine and a second of the different sub-expressions is directed to the general filter sub-engine.

14. (Currently Amended) The filter engine system as recited in claim 8, wherein the optimized filter sub-engine comprises:

a first optimized filter sub-engine configured to process inputs that conform to a first subset of the language; and

a second optimized filter sub-engine configured to process inputs that conform to a second subset of the language;

wherein the first subset of the language is different from the second subset of the input language.

15. **(Currently Amended)** A computer-readable storage medium encoded with instructions that, when executed by a processor of a device, cause the device to perform acts comprising:

determining an appropriate filter sub-engine to which an input message should be directed for processing against a set of queries;

processing the input message using an optimized filter sub-engine if the optimized filter sub-engine comprises a grammar that supports processing of the input message;

processing the input message in a general filter sub-engine if the optimized filter sub-engine grammar does not support processing of the input message; and

wherein:

the input message is in accordance with a query language based on eXtensible Markup Language (XML);

the optimized filter sub-engine supports a subset, less than the whole, of the query language; and

the general filter sub-engine supports the entire query language.

16. **(Previously Presented)** The computer-readable storage medium as recited in claim 15, further comprising computer-executable instructions that, when executed, direct the computing system to perform acts comprising:

accept input messages for both the optimized filter sub-engine and the general filter sub-engine by way of a single input means so that an input message sending application is not required to distinguish between the optimized filter sub-engine and the general filter sub-engine.

17. **(Previously Presented)** The computer-readable storage medium as recited in claim 15, wherein the query language is XPath.

18. **(Previously Presented)** The computer-readable storage medium as recited in claim 15, wherein the query language is an XML query language.

19. **(Previously Presented)** The computer-readable storage medium as recited in claim 15, further comprising computer-executable instructions that, when executed, direct the computing system to perform acts comprising:

prior to determining which filter sub-engine will process the input message, parse the input message into two or more sub-expressions;

for each of the two or more sub-expressions, determine an appropriate filter sub-engine that can process the sub-expression; and

direct each of the two or more sub-expressions to the appropriate filter sub-engine for processing.

20. **(Previously Presented)** The computer-readable storage medium as recited in claim 19, further comprising computer-executable instructions that, when executed, direct the computing system to derive a final result of the input message processing from at least one result of the sub-expression processing.

21. (Previously Presented) The computer-readable storage medium as recited in claim 19, further comprising computer-executable instructions that, when executed, direct the computing system to perform acts comprising:

determine if a first of the two or more sub-expressions evaluates true;

proceed with processing of subsequent sub-expressions of the two or more sub-expressions if the first sub-expression evaluates to true; and

forego processing of subsequent sub-expressions of the two or more sub-expressions if the first sub-expression evaluates to false.

22. (Previously Presented) The computer-readable storage medium as recited in claim 15, wherein each filter sub-engine includes a set of queries against which input messages directed to the respective filter sub-engine are tried, and wherein each set of queries is unique.

23. – 32. (Canceled)

33. (Previously Presented) The method as recited in claim 1, wherein: determining comprises generating a hash of the input data in order to determine if an optimized sub-engine is capable of handling the input data.